S25-63
118
188194

# Adaptive Hybrid Control of Manipulators

H. Seraji
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109

JJ57049

**Abstract**

~~The paper presents~~ simple methods for the design of adaptive force and position controllers for robot manipulators within the hybrid control architecture ~~. The~~ is presented. force controller is composed of an adaptive PID feedback controller, an auxiliary signal and a force feedforward term; and it achieves tracking of desired force setpoints in the constraint directions. The position controller consists of adaptive feedback and feedforward controllers and an auxiliary signal; and it accomplishes tracking of desired position trajectories in the free directions. The controllers are capable of compensating for dynamic cross-couplings that exist between the position and force control loops in the hybrid control architecture. The adaptive controllers do not require knowledge of the complex dynamic model or parameter values of the manipulator or the environment. The proposed control schemes are computationally fast and suitable for implementation in on-line control with high sampling rates.

## 1. Introduction

Although control of robot manipulators has been studied extensively in recent years, this study has been focused primarily on position control of manipulators in free motion within an unconstrained environment. In many practical applications, the manipulator is constrained by the environment and certain degrees-of-freedom are lost for motion due to environmental constraints. When the manipulator makes contact with the environment, the contact forces must be controlled in the constraint directions, while the positions are controlled simultaneously in the free directions.

The problem of manipulator control in a constrained environment has been investigated by several researchers [1]. At present, three major conceptual approaches exist for simultaneous position and force control. Paul and Shimano [2] suggest a method which uses certain joints for position control while the remaining joints are used for force control. Salisbury [3] puts forward a technique for controlling the end-effector stiffness characteristics in the Cartesian space. Raibert and Craig [4] propose a conceptual architecture, based on the analysis of Mason [5], for "hybrid control" which allows forces to be controlled in the constraint directions by a force controller, while simultaneously controlling positions in the free directions by a position controller. Raibert and Craig, however, do not prescribe a general and systematic method for the design of position and force controllers. Nevertheless, hybrid control has gained considerable popularity over the other two alternatives for simultaneous position and force control [6-13].

The present paper puts forth systematic methods for the design of adaptive force and position controllers within the hybrid control architecture. The force controller achieves tracking of desired force setpoints, while the position controller accomplishes tracking of desired position trajectories. The force and position controller gains are generated by adaptation laws by means of simple arithmetic operations, and thus the controllers are computationally fast and suitable for on-line implementation with high sampling rates. The adaptive controllers do not require knowledge of the complex dynamic model or parameter values of the manipulator or the environment.

The paper is structured as follows. In Section 2, the hybrid control architecture is outlined and the problem is stated. Section 3 addresses the design of force control system using model reference adaptive control (MRAC) theory. The design of position control system is discussed briefly in Section 4. In Section 5, the force and position controllers are integrated in the hybrid control architecture. Finally, Section 6 discusses the results of the paper and draws some conclusions.

## 2. Problem Statement

In this section, the hybrid force/position* control architecture is discussed briefly, and the force and position tracking control problems are stated.

Let us consider a robot manipulator which performs a number of different tasks in a Cartesian space {X}. Each task, in general, involves motion of the manipulator end-effector in certain directions and, simultaneously, exertion of force by the end-effector on the environment in the remaining directions [5]. The directions of motion and force depend on the nature of the particular task to be performed and are reflected in the "task matrix" in the hybrid control architecture shown in Figure 1. The task matrix also contains the transformations required to map the measurements $\underline{\theta}$ and $\underline{F}_s$ of joint encoders and force/torque sensors into position and force variables in the constraint frame defined with respect to the task geometry [4]. Note that the desired force and position trajectories are also specified in the constraint frame. For any given task, the n-dimensional Cartesian space {X} can be decomposed into two orthogonal $\ell$- and m-dimensional subspaces {Y} and {Z}, where $n = \ell + m$. The "position subspace" {Y} contains the $\ell$ directions (i.e., degrees-of-freedom) in which the manipulator end-effector is free to move and along which end-effector position is to be controlled. The "force subspace" {Z} contains the remaining m directions in which the manipulator end-effector is constrained by and interacts with the environment and along which the contact force is to be controlled.

In the hybrid force/position control problem addressed in this paper, we consider the "virtual" Cartesian force $\underline{F}$ acting on the end-effector as the manipulated variable and the position or force of the end-effector as the controlled variables [14]. The hybrid control architecture is based on two independent and non-interacting controllers as shown in Figure 1; namely, the position controller which operates in {Y} and the force controller which acts in {Z}. The position controller generates the Cartesian end-effector force $\underline{F}_y$ required to cause the end-effector motion to track a desired position trajectory in {Y}. The force controller produces the Cartesian end-effector force $\underline{F}_z$ needed to ensure that the end-effector force follows a desired force setpoint in {Z}. Since we cannot physically apply Cartesian forces to the end-effector, we instead compute and implement the equivalent joint torques needed to effectively cause these forces. The required joint torques are obtained from the Cartesian forces by means of the Jacobian matrix $J(\underline{\theta})$ of the manipulator, where $\underline{\theta}$ is the joint angle vector.

We shall now address the problems of force and position control separately in Sections 3 and 4 and then integrate the results in Section 5.

## 3. Design of Force Control System

In this section, a simple dynamic model for force control in the subspace {Z} is described, and an adaptive force control scheme is developed.

### 3.1 Dynamic Force Model

The full dynamic model of the end-effector plus force/torque sensor in contact with the environment is complex [15]. However, the dynamic behavior of this system can be modelled approximately by a mass-spring-damper in each degree-of-freedom as shown in Figure 2 and described by the differential equation

$$m\,\ddot{z}(t) + d\,\dot{z}(t) + k\,z(t) = f(t) \tag{1}$$

Generalizing this simple model to the m-dimensional force subspace {Z}, the dynamic behavior of the system in {Z} can be expressed by the differential equation

$$M_o\ddot{\underline{Z}}(t) + D_o\dot{\underline{Z}}(t) + K_e\underline{Z}(t) = \underline{F}_z(t) \tag{2}$$

where $\underline{Z}(t)$ is the mx1 end-effector position/orientation vector, $M_o$ is the symmetric positive-definite mxm generalized mass matrix, $D_o$ is the mxm generalized damping matrix, $K_e$ is the diagonal mxm generalized stiffness matrix and $\underline{F}_z$ is the mx1 force vector applied to the end-effector in the force subspace {Z}. The elements of $K_e$ are the "equivalent" translational (force) and rotational (torque) coefficients of elasticity (stiffness) of the system in various directions in {Z}. By an appropriate choice of the {Z} subspace origin, the mx1 force/torque vector $\underline{P}(t)$ exerted by the end-effector on the environment is related to $\underline{Z}(t)$ by the generalization of Hooke's law as

$$\underline{P}(t) = K_e\,\underline{Z}(t) \tag{3}$$

From equations (2) and (3), we obtain

$$A\,\ddot{\underline{P}}(t) + B\,\dot{\underline{P}}(t) + \underline{P}(t) = \underline{F}_z(t) \tag{4}$$

where $A = M_oK_e^{-1}$ and $B = D_oK_e^{-1}$ are mxm matrices. Equation (4) gives a simple dynamic model of the system in the force subspace {Z}. Since the manipulator dynamics is highly nonlinear, the matrices A and B in equation (4) are dependent on the end-effector Cartesian position and velocity vectors $\underline{X}$ and $\dot{\underline{X}}$ and also on the system parameters such as the equivalent stiffness and the payload mass, which are represented by the parameter vector $\underline{p}$. Furthermore, due to internal cross-coupling of the manipulator dynamics, a "disturbance" term $\underline{C}_p(\underline{Y})$ must be included in equation (4) to represent the dynamic coupling from the position loop into the force loop, where $\underline{Y}$

---

* In this paper, "position" implies position and orientation and "force" implies force and torque.

is the end-effector position vector in (Y). Thus, a more realistic model for force control is obtained as

$$A(\underline{Y},\dot{\underline{Y}},\underline{p})\ \ddot{\underline{P}}(t) + B(\underline{Y},\dot{\underline{Y}},\underline{p})\dot{\underline{P}}(t) + \underline{P}(t) + \underline{C}_p(\underline{Y}) = \underline{P}_s(t) \qquad (5)$$

Equation (5) is a set of highly complex nonlinear and coupled second-order differential equations.

### 3.2 Force Control Scheme

In order to control the force/torque $\underline{P}(t)$ exerted by the end-effector on the environment, let us employ a PID controller with adaptive gains $\{K_p(t),\ K_I(t),\ K_D(t)\}$ and an auxiliary signal $\underline{d}(t)$ in the force control law

$$\underline{P}_s(t) = \underline{P}_r(t) + K_p(t)\ \underline{E}(t) + K_I(t) \int_0^t \underline{E}(t)dt + K_D(t)\ \dot{\underline{E}}(t) + \underline{d}(t) \qquad (6)$$

where $\underline{P}_r(t)$ is the mx1 vector of desired force trajectory used as a feedforward term, and the mx1 force tracking-error vector $\underline{E}(t) = \underline{P}_r(t) - \underline{P}(t)$ is the deviation of the actual (measured) force from the desired value. Since in practical applications the desired force trajectory is very often a constant setpoint $\underline{P}_r(t) = \underline{P}_r$, the PID control law is particularly suitable for this situation. Furthermore, the auxiliary signal $\underline{d}(t)$ compensates for the cross-coupling term $\underline{C}_p$ and the time and parameter variations of A and B matrices. Note that the feedforward term $\underline{P}_r(t)$ is included in the control law (6) since ideally we want $\underline{P}(t) = \underline{P}_r(t)$. In equation (6), the gains of the PID controller, namely $K_p(t)$, $K_I(t)$ and $K_D(t)$, and also the auxiliary signal $\underline{d}(t)$ are adapted in real-time to accomplish force setpoint tracking in spite of the nonlinear and possibly time-varying behavior of the system model (5).

On applying the linear control law (6) to the system model (5), we obtain

$$A\ \ddot{\underline{P}}(t) + B\ \dot{\underline{P}}(t) + \underline{P}(t) + \underline{C}_p = \underline{P}_r(t) + K_p\underline{E}(t) + K_I \int_0^t \underline{E}(t)dt + K_D\dot{\underline{E}}(t) + \underline{d}(t) \qquad (7)$$

Using $\underline{E}(t) = \underline{P}_r(t) - \underline{P}(t)$ and noting that $\dot{\underline{E}}(t) = -\dot{\underline{P}}(t)$ and $\ddot{\underline{E}}(t) = -\ddot{\underline{P}}(t)$ for a constant desired force, equation (7) can be written as

$$\ddot{\underline{E}}(t) + A^{-1}(B + K_D)\ \dot{\underline{E}}(t) + A^{-1}(I + K_p)\ \underline{E}(t) + A^{-1}K_I\ \underline{E}^*(t) = A^{-1}\ [\underline{C}_p - \underline{d}(t)] \qquad (8)$$

where $\underline{E}^*(t) = \int_0^t \underline{E}(t)dt$ is the mx1 integral error vector. Equation (8) can be expressed in standard state-space form as

$$\frac{d\underline{x}_e(t)}{dt} = \begin{pmatrix} 0 & I_m & 0 \\ 0 & 0 & I_m \\ -A^{-1}K_I & -A^{-1}(I+K_p) & -A^{-1}(B+K_D) \end{pmatrix} \underline{x}_e(t) + \begin{pmatrix} 0 \\ 0 \\ A^{-1}(\underline{C}_p-\underline{d}) \end{pmatrix} \qquad (9)$$

where $\underline{x}_e(t) = \begin{pmatrix} \underline{E}^*(t) \\ \underline{E}(t) \\ \dot{\underline{E}}(t) \end{pmatrix}$

is the 3mx1 augmented error vector. Equation (9) constitutes the "adjustable system" in the MRAC framework.

Now, in the ideal situation, the desired behavior of the force error $\underline{E}_m(t)$ is described by the homogeneous differential equation

$$\ddot{\underline{E}}_m(t) + D_3\dot{\underline{E}}_m(t) + D_2\underline{E}_m(t) + D_1\underline{E}_m^*(t) = \underline{0} \qquad (10)$$

where $D_1$, $D_2$ and $D_3$ are constant mxm matrices which are chosen such that equation (10) is stable and embodies the desired performance of the force control system. By choosing $D_1$, $D_2$ and $D_3$ as diagonal matrices, the force errors will be decoupled; for instance

$$\ddot{E}_{mi}(t) + d_{3i}\dot{E}_{mi}(t) + d_{2i}E_{mi}(t) + d_{1i}E_{mi}^*(t) = 0 \qquad (11)$$

where the coefficients $d_{1i}$, $d_{2i}$ and $d_{3i}$ are chosen such that the tracking-error $E_i(t) = P_{ri}(t) - P_i(t)$ has a desired behavior and $d_{2i}d_{3i} > d_{1i}$ to ensure stability. Equation (10) can be written as

$$\dot{\underline{x}}_m(t) = \begin{pmatrix} 0 & I_m & 0 \\ 0 & 0 & I_m \\ -D_1 & -D_2 & -D_3 \end{pmatrix} \underline{x}_m(t) = D\ \underline{x}_m(t) \qquad (12)$$

where $\underline{x}_m(t) = \begin{pmatrix} \underline{E}_m^*(t) \\ \underline{E}_m(t) \\ \dot{\underline{E}}_m(t) \end{pmatrix}$ is the 3mx1 desired error vector. Equation (12) constitutes the "reference model" in the context of MRAC theory. Since the initial values of the actual and desired forces are often the same, the initial error $\underline{x}_m(0)$ is equal to zero, and hence from equation (12), $\underline{x}_m(t) = \exp[Dt] \cdot \underline{x}_m(0) \equiv \underline{0}$ for all t.

Now, in order for the adjustable system state $z_e(t)$ to tend to the reference model state $z_m(t) = 0$ asymptotically, from Reference [16] we require

$$\begin{pmatrix} 0 \\ 0 \\ A^{-1}\dot{d} \end{pmatrix} = Q_o^{-1} M z_e + Q_o^{*} M \dot{z}_e \qquad (13a)$$

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ A^{-1}\dot{K}_I & A^{-1}\dot{K}_P & A^{-1}\dot{K}_D \end{pmatrix} = Q_1^{-1} M z_e z_e' \begin{pmatrix} \alpha_1 I_m & 0 & 0 \\ 0 & \beta_1 I_m & 0 \\ 0 & 0 & \gamma_1 L \end{pmatrix}$$

$$+ Q_1^{*} M \frac{d}{dt}(z_e z_e') \begin{pmatrix} \alpha_2 I_m & 0 & 0 \\ 0 & \beta_2 I_m & 0 \\ 0 & 0 & \gamma_2 L \end{pmatrix} \qquad (13b)$$

where " ' " denotes transposition, $\{\alpha_1, \beta_1, \gamma_1\}$ are positive scalars, $\{\alpha_2, \beta_2, \gamma_2\}$ are zero or positive scalars, and $L$ is an $m\times m$ constant matrix to be specified later. In equation (13), $Q_o$ and $Q_1$ are symmetric positive-definite $3m\times 3m$ matrices, $Q_o^*$ and $Q_1^*$ are symmetric positive semi-definite $3m\times 3m$ matrices, and the symmetric positive-definite $3m\times 3m$ matrix

$$M = \begin{pmatrix} M_1 & M_2 & M_3 \\ M_2 & M_4 & M_5 \\ M_3 & M_5 & M_6 \end{pmatrix}$$

is the solution of the Lyapunov equation for the reference model (12), namely

$$M D + D' M = -N \qquad (14)$$

where $N$ is a symmetric positive-definite $3m\times 3m$ matrix. In deriving equation (13), the matrices A, B, and $C_p$ in the robot model (5) are assumed to be unknown and "slowly time-varying" compared with the adaptation algorithm, since these matrices cannot change significantly in each sampling interval which is of the order of a millisecond. Now, in order to make the controller adaptation laws <u>independent</u> of the model matrix A, we choose

$$Q_o = \frac{1}{\delta_1} A^* \;\; ; \;\; Q_1 = A^* \;\; ; \;\; Q_o^* = \delta_2 [A^*]^{-1} \;\; ; \;\; Q_1^* = [A^*]^{-1} \qquad (15)$$

where $\{\delta_1, \delta_2\}$ are positive and zero or positive scalars, and

$A^* = \begin{pmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{pmatrix}$ is a symmetric positive-definite $3m\times 3m$ matrix.

Substituting from equation (15) into equation (13), after simplification we obtain the adaptation laws

$$\dot{d}(t) = \delta_1 q(t) + \delta_2 \dot{q}(t) \qquad (16)$$

$$\dot{K}_I(t) = \alpha_1 [q(t)\, E^{*\prime}(t)] + \alpha_2 \frac{d}{dt}[q(t)\, E^{*\prime}(t)] \qquad (17)$$

$$\dot{K}_P(t) = \beta_1 [q(t)\, E'(t)] + \beta_2 \frac{d}{dt}[q(t)\, E'(t)] \qquad (18)$$

$$\dot{K}_D(t) = \gamma_1 [q(t)\, \dot{E}'(t) L] + \gamma_2 \frac{d}{dt}[q(t)\, \dot{E}'(t) L] \qquad (19)$$

where $q(t)$ is the $m\times 1$ "weighted" force error vector defined as

$$q(t) = M_3\, E^*(t) + M_5\, E(t) + M_6\, \dot{E}(t) \qquad (20)$$

and $M_3$, $M_5$, $M_6$ are appropriate submatrices of M. Thus, the required auxiliary signal and PID controller gains are obtained as

$$d(t) = d(0) + \delta_1 \int_0^t q(t)dt + \delta_2 q(t) \qquad (21)$$

$$K_I(t) = K_I(0) + \alpha_1 \int_0^t q(t)\, E^{*\prime}(t)dt + \alpha_2 q(t)\, E^{*\prime}(t) \qquad (22)$$

$$K_p(t) = K_p(0) + \beta_1 \int_0^t \underline{q}(t)\, \underline{E}'(t)dt + \beta_2 \underline{q}(t)\, \underline{E}'(t) \tag{23}$$

$$K_D(t) = K_D(0) + \gamma_1 \int_0^t \underline{q}(t)\, \dot{\underline{E}}'(t)L dt + \gamma_2 \underline{q}(t)\, \dot{\underline{E}}'(t)L \tag{24}$$

The force control law is then given by

$$\underline{P}_s(t) = \underline{P}_r(t) + \underline{d}(t) + K_I(t) \int_0^t \underline{E}(t)dt + K_p(t)\underline{E}(t) + K_D(t)\dot{\underline{E}}(t) \tag{25}$$

It is noted that the auxiliary signal $\underline{d}(t)$ can be generated by a $PI^2D$ controller driven by the force error $\underline{E}(t)$ since, from equations (20)-(21), $\underline{d}(t)$ can be expressed as

$$\underline{d}(t) = \underline{d}(0) + [\delta_2 N_6]\dot{\underline{E}}(t) + [\delta_1 N_6 + \delta_2 N_5]\underline{E}(t)$$

$$+ [\delta_1 N_5 + \delta_2 N_3] \int_0^t \underline{E}(t)dt + [\delta_1 N_3] \int_0^t \left\{ \int_0^t \underline{E}(t)dt \right\} dt$$

In practical implementation of any force control law, differentiation of the noisy force measurement $\underline{P}(t)$ is undesirable and, moreover, differentiation of the constant force setpoint $\underline{P}_r$ produces unwanted impulses. This argument suggests that the derivative $\dot{\underline{E}}(t)$ in equations (20), (24) and (25) must be replaced by $-K_e\dot{\underline{Z}}(t)$ using equation (3). This yields the linear adaptive force control law

$$\underline{P}_s(t) = \underline{P}_r(t) + \underline{d}(t) + K_I(t) \int_0^t \underline{E}(t)dt + K_p(t)\underline{E}(t) - K_v(t)\dot{\underline{Z}}(t) \tag{26}$$

which is shown in Figure 3, where $K_v(t) = K_D(t)K_e$ is the mxm velocity feedback gain matrix and the term $K_v(t)\dot{\underline{Z}}(t)$ represents velocity damping. Choosing $L = (K_e^{-1})^2$, the adaptation laws now become

$$\underline{d}(t) = \underline{d}(0) + \delta_1 \int_0^t \underline{q}(t)dt + \delta_2 \underline{q}(t) \tag{27}$$

$$K_I(t) = K_I(0) + \alpha_1 \int_0^t \underline{q}(t)\underline{E}^{*\prime}(t)dt + \alpha_2 \underline{q}(t)\underline{E}^{*\prime}(t) \tag{28}$$

$$K_p(t) = K_p(0) + \beta_1 \int_0^t \underline{q}(t)\underline{E}'(t)dt + \beta_2 \underline{q}(t)\underline{E}'(t) \tag{29}$$

$$K_v(t) = K_v(0) - \gamma_1 \int_0^t \underline{q}(t)\dot{\underline{Z}}'(t)dt - \gamma_2 \underline{q}(t)\dot{\underline{Z}}'(t) \tag{30}$$

where

$$\underline{q}(t) = N_3\underline{E}^*(t) + N_5\underline{E}(t) - N_6^*\dot{\underline{Z}}(t) \tag{31}$$

and $N_6^* = N_6 K_e$. It can be shown that by proper selection of matrix N in the Lyapunov equation (14), the submatrices $N_3$, $N_5$ and $N_6$ in equation (20) can be made equal to the desired values $W_I$, $W_p$ and $W_D = W_v K_e^{-1}$ respectively, and hence equation (31) becomes

$$\underline{q}(t) = W_I\underline{E}^*(t) + W_p\underline{E}(t) - W_v\dot{\underline{Z}}(t) \tag{32}$$

where $W_I$, $W_p$ and $W_v$ are the mxm diagonal weighting matrices chosen by the designer to reflect the relative significance of the integral error $\underline{E}^*$, the force error $\underline{E}$, and the velocity $\dot{\underline{Z}}$, respectively. From equations (27)-(30), the general expression for a typical controller gain $K(t)$ which acts on the signal $\underline{y}(t)$ to generate the term $K(t)\underline{y}(t)$ in the control law (26) can be written as

$$K(t) = K(0) + \mu_1 \int_0^t \underline{q}(t)\underline{y}'(t)dt + \mu_2 \underline{q}(t)\underline{y}'(t) \tag{33}$$

where $\mu_1$ and $\mu_2$ are scalar gains. This computation can be performed by a simple "adaptation module" shown in the block diagram of Figure 4 (For $\underline{d}(t)$, we set $\underline{y} = 1$). The adaptation module acts on the two input signals $\underline{q}(t)$ and $\underline{y}(t)$ to produce the output signal $K(t)\underline{y}(t)$. The force control law (26) can then be constructed by parallel connection of four such modules.

The force control scheme developed in this section is extremely simple, since the adaptation laws (27)-(30) generate the controller gains by means of simple integration using, for instance, the trapezoidal rule in equation (33) can be implemented as

$$K(i) = K(i-1) + \mu_1 \cdot \frac{T_s}{2} [q(i)r'(i) + q(i-1)r'(i-1)] + \mu_2[q(i)r'(i)] \qquad (34)$$

where the integer $i$ denotes the sampling instant and $T_s$ is the sampling period. As a result, the force control law (26) can be evaluated very rapidly and consequently, the force control scheme can be implemented for real-time control with high sampling rates (typically 1 KHz). High sampling rate is very desirable in force control applications and yields improved dynamic performance. The adaptation laws (27)-(30) do not require the complex nonlinear model of manipulator dynamics (5) or any knowledge of parameters of the manipulator or the environment. This is due to the fact that the adaptive force controller has "learning capabilities" and can rapidly adapt itself to gross changes in the manipulator or the environment parameters.

## 4. Design of Position Control System

In this section, a dynamic model for position control in the subspace {Y} is described and an adaptive position control scheme is briefly explained.

### 4.1 Dynamic Position Model

The dynamics of the end-effector in the Cartesian space {X} can be represented by [14]

$$\Lambda(X)\ddot{X} + \mu(X, \dot{X}) + p(X) + P = F \qquad (35)$$

where $\Lambda$ is the Cartesian mass matrix, $\mu$ is the Cartesian centrifugal, Coriolis and friction vector, $p$ is the Cartesian gravity loading vector, $P$ is the force vector exerted by the end-effector on the environment, and $F$ is the generalized "virtual" Cartesian force vector applied to the end-effector. In the position subspace {Y}, equation (35) can be written as [17]

$$A(X, \dot{X}, p)\ \ddot{Y}(t) + B(X, \dot{X}, p)\ \dot{Y}(t) + C_o(X, \dot{X}, p)\ Y(t) + C_f(P) = F_y(t) \qquad (36)$$

where the $\ell x \ell$ matrices A, B, $C_o$ are highly complex nonlinear functions of $X$, $\dot{X}$ and the system parameters $p$. $C_f$ represents the dynamic coupling effect from the force loop into the position loop which is a function of the force vector $P$ in {Z}, and $F_y$ is the $\ell$x1 force vector applied to the end-effector in the position subspace {Y}. Equation (36) is a set of highly complex nonlinear and coupled second-order differential equations.

### 4.2 Position Control Scheme

The Cartesian position control scheme is developed fully in Reference [18]. For the sake of completeness, the results are summarized in this section.

The linear adaptive position control law is given by

$$F_y(t) = f(t) + K_p(t)E_p(t) + K_v(t)\dot{E}_p(t) + C(t)R(t) + B(t)\dot{R}(t) + A(t)\ddot{R}(t) \qquad (37)$$

as shown in Figure 5, where $R(t)$ is the $\ell$x1 reference (desired) position trajectory vector, $E_p(t) = R(t) - Y(t)$ is the $\ell$x1 position tracking-error vector, $f(t)$ is an auxiliary signal, and $[K_pE_p + K_v\dot{E}_p]$ and $[CR + B\dot{R} + A\ddot{R}]$ are the contributions due to the feedback and feedforward controllers respectively. The required auxiliary signal and controller gains are adapted according to the following laws:

$$f(t) = f(0) + \delta_1 \int_0^t r(t)dt + \delta_2 r(t) \qquad (38)$$

$$K_p(t) = K_p(0) + \nu_1 \int_0^t r(t)E_p'(t)dt + \nu_2 r(t)E_p'(t) \qquad (39)$$

$$K_v(t) = K_v(0) + \eta_1 \int_0^t r(t)\dot{E}_p'(t)dt + \eta_2 r(t)\dot{E}_p'(t) \qquad (40)$$

$$C(t) = C(0) + \mu_1 \int_0^t r(t)R'(t)dt + \mu_2 r(t)R'(t) \qquad (41)$$

$$B(t) = B(0) + \gamma_1 \int_0^t r(t)\dot{R}'(t)dt + \gamma_2 r(t)\dot{R}'(t) \qquad (42)$$

$$A(t) = A(0) + \lambda_1 \int_0^t r(t)\ddot{R}(t)dt + \lambda_2 r(t)\ddot{R}(t) \qquad (43)$$

where the $\ell$x1 "weighted" position error vector $r(t)$ is defined as

$$\underline{x}(t) = W_p \underline{E}_p(t) + W_v \dot{\underline{E}}_p(t) \qquad (44)$$

In equations (38)-(43), $\{\delta_1, \nu_1, \eta_1, \mu_1, \gamma_1, \lambda_1\}$ are positive scalars, $\{\delta_2, \nu_2, \eta_2, \mu_2, \gamma_2, \lambda_2\}$ are positive or zero scalars, and $W_p$ and $W_v$ are weighting matrices specified by the designer to reflect the relative significance of the position and velocity errors $\underline{E}_p$ and $\dot{\underline{E}}_p$. Note that from equations (38)-(43), the controller gains can be computed using the same adaptation "module" as in Section 4 (eqn. 33) shown in Figure 4. It is seen that the position control scheme is extremely simple since the controller gains are obtained from equations (38)-(43) by simple integration (such as trapezoidal rule) and thus the computational time required to evaluate the position control law (37) is extremely short. Thus, the position control scheme can be implemented for on-line control with high sampling rates (~ 1 KHz), resulting in improved dynamic performance.

## 5. Hybrid Force/Position Control System

In Sections 3 and 4, the Cartesian end-effector forces $\underline{F}_z$ and $\underline{F}_y$ are generated by the force and position controllers to accomplish force and position tracking in {Z} and {Y} respectively. Since Cartesian forces cannot be applied to the end-effector in practice, these end-effector forces must be mapped into the equivalent joint torques. Thus, in order to implement the force and position controllers, the control law in joint space is given by [19]

$$\underline{T}(t) = J'(\underline{\theta}) \begin{pmatrix} \underline{F}_z(t) \\ \underline{F}_y(t) \end{pmatrix} \qquad (45)$$

where $\underline{\theta}$ is the nx1 joint angle vector, $\underline{T}$ is the nx1 joint torque vector, and $J$ is the nxn Jacobian matrix of the manipulator, with appropriate reordering of the columns of $J$ if necessary.

It is important to note that although the force and position controllers are separate in the hybrid control architecture, there exists dynamic cross-coupling from the force control loop into the position control loop and vice versa. This coupling is due to the fact that the end-effector dynamics in the Cartesian space {X} is strongly cross-coupled; i.e. the application of end-effector force in any direction affects the end-effector positions in all directions. The cross-coupling effects are modelled as "disturbance" terms $\underline{\zeta}_p$ and $\underline{\zeta}_f$ in the force and position control loops. The adaptive controllers are capable of compensating for these disturbances and maintaining a good tracking performance. The ability to cope with cross-coupling effects in the hybrid control architecture is an important feature of the adaptive control schemes of Sections 3 and 4.

## 6. Discussion and Conclusions

Simple adaptive force and position control schemes for manipulators in a hybrid control architecture are described in this paper. The control schemes are computationally fast and do not require the complex dynamic model or parameter values of the manipulator or the environment. The force and position control loops are stable since the design is based on the Lyapunov method which guarantees stability as a by-product of the design.

There are certain differences between the proposed approach and the conventional hybrid control of Raibert and Craig [4]. Firstly, in the present approach, the force and position control problems are formulated in the Cartesian space with the end-effector Cartesian forces as the manipulated variables; whereas in [4], the problems are formulated in the joint space. The proposed formulation results in computational improvement since inverse Jacobians are not needed in the control loops. Secondly, in the proposed approach, the "task matrix" operates on the measured variables so as to produce the position and force variables that need to be controlled; whereas in [4], a selection matrix and its complement are used after formation of tracking-errors. The present approach seems more straightforward and appealing than the conventional approach.

An attractive feature of the adaptive controllers designed in this paper is their abilities to compensate for dynamic cross-couplings that exist between the position and force control loops in the hybrid control architecture. Furthermore, the adaptive force and position controllers have "learning capabilities" to cope with unpredictable changes in the manipulator or environment parameters such as the stiffness This is due to the fact that the controller gains are adapted rapidly on the basis of the manipulator performance. The low computational requirements make the proposed control schemes suitable for implementation in on-line hybrid control with high sampling rates.

## 7. Acknowledgement

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

## 8. References

[1] D.E. Whitney: "Historical perspective and state of the art in robot force control," Proc. IEEE Intern. Conf. on Robotics and Automation, pp. 262-268, St. Louis, MO, 1985.

[2] R.P. Paul and B. Shimano: "Compliance and control," Proc. Joint Aut. Control Conf., pp. 694-699, San Francisco, CA, 1976.

[3] J.K. Salisbury: "Active stiffness control of a manipulator in Cartesian coordinates," Proc. IEEE Conf. on Decision and Control, pp. 95-100, Albuquerque, NM, 1980.

[4] M.H. Raibert and J.J. Craig: "Hybrid position/force control of manipulators," ASME J. Dyn. Systems, Measurement and Control, Vol. 102, pp. 126-133, 1981.

[5] M.T. Mason: "Compliance and force control for computer controlled manipulators," IEEE Trans. Systems, Man and Cybernetics, SMC11(6), pp. 418-432, 1981.

[6] H. Zhang and R.P. Paul: "Hybrid control of robot manipulators," Proc. IEEE Intern. Conf. on Robotics and Automation, pp. 602-607, St. Louis, MO, 1985.

[7] R. West and H. Asada: "A method for the design of hybrid position/force controllers for manipulators constrained by contact with the environment," Proc. IEEE Intern. Conf. on Robotics and Automation, pp. 251-259, St. Louis, MO, 1985.

[8] P.G. Backes, G.G. Leininger and C.H. Chung: "Real-time Cartesian coordinate hybrid control of a PUMA 560 manipulator," Proc. IEEE Intern. Conf. on Robotics and Automation, pp. 608-613, St. Louis, MO, 1985.

[9] P.G. Backes, G.G. Leininger and C.H. Chung: "Joint self-tuning with Cartesian setpoints," ASME J. Dyn. Systems, Measurement and Control, Vol. 108, pp. 146-150, 1986.

[10] O. Khatib and J. Burdick: "Motion and force control of robot manipulators," Proc. IEEE Intern. Conf. on Robotics and Automation, pp. 1381-1386, San Francisco, CA, 1986.

[11] T. Yoshikawa: "Dynamic hybrid position/force control of robot manipulators," Proc. IEEE Intern. Conf. on Robotics and Automation, pp. 1393-1398, San Francisco, CA, 1986.

[12] A.J. Koivo: "Force-position-velocity control with self-tuning for robotic manipulators," Proc. IEEE Intern. Conf. on Robotics and Automation, pp. 1563-1568, San Francisco, CA, 1986.

[13] C.C. Nguyen, F.J. Pooran and T. Premack: "Control of robot manipulator compliance," Proc. Intern. Symp. on Robot Manipulators, Albuquerque, NM, 1986.

[14] O. Khatib: "Dynamic control of manipulators in operational space," Proc. 6th IFToMM Congress on Theory of Machines and Mechanisms, pp. 1128-1131, New Delhi, India, 1983 (Wiley, New Delhi).

[15] S.D. Eppinger and W.P. Seering: "On dynamic models of robot force control," Proc. IEEE Intern. Conf. on Robotics and Automation, pp. 29-34, San Francisco, CA, 1986.

[16] H. Seraji: "A simple method for model reference adaptive control," submitted for publication to IEEE Trans. Aut. Control, 1986.

[17] A. Balestrino, G. DeMaria and L. Sciavicco: "An adaptive model following control for robotic manipulators," ASME J. Dyn. Systems, Measurement and Control, Vol. 105, pp. 143-151, 1983.

[18] H. Seraji: "Direct adaptive control of manipulators in Cartesian space," Journal of Robotic Systems, Vol. 4, No. 1, pp. 157-178, 1987.

[19] J.J. Craig: "Robotics-Mechanics and Control," Addison-Wesley Publishing Company, Reading, MA, 1986.

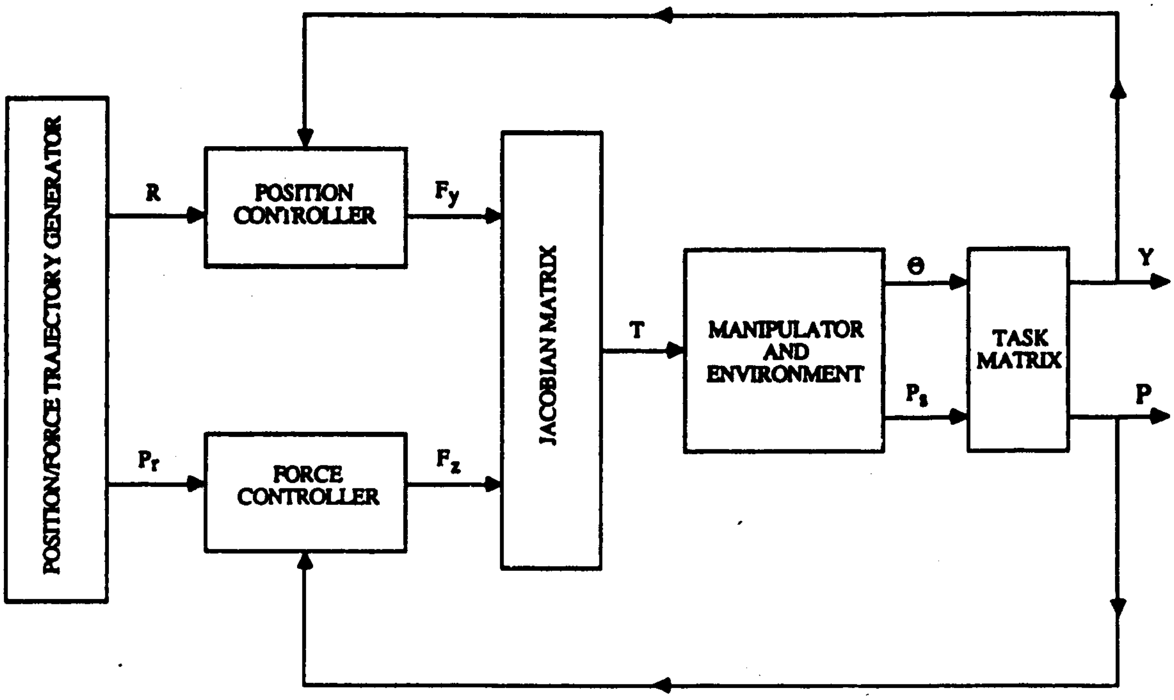


Figure 1. Hybrid Control Architecture

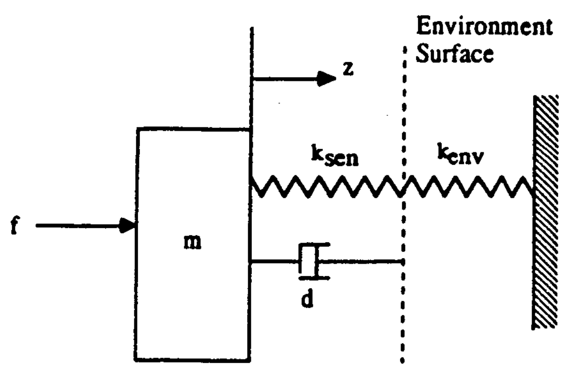


Figure 2. Mass-Spring-Damper Model

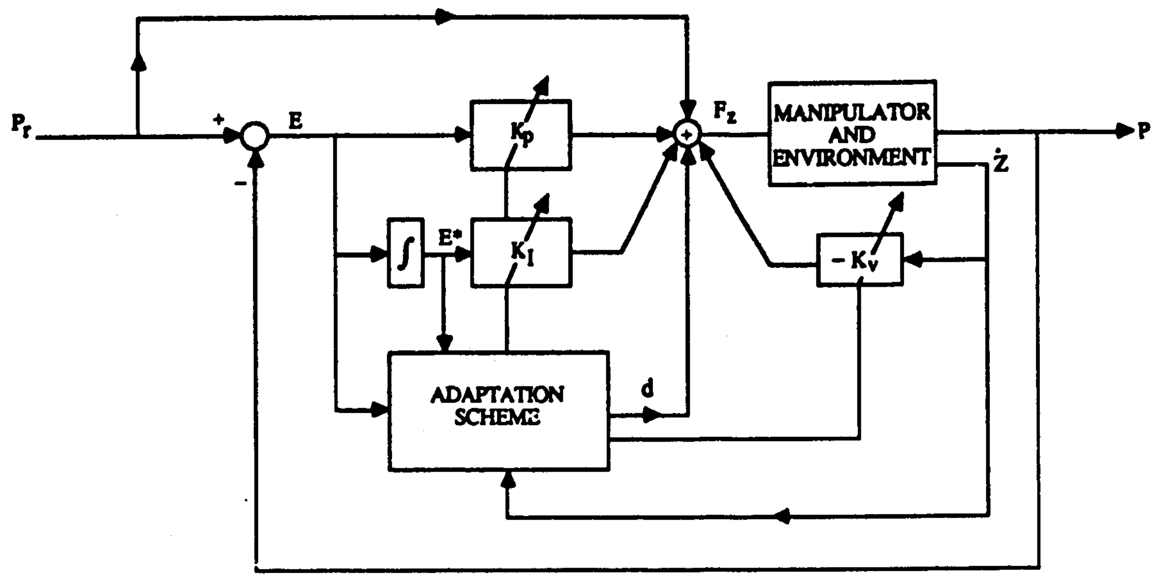


Figure 3. Adaptive Force Control System

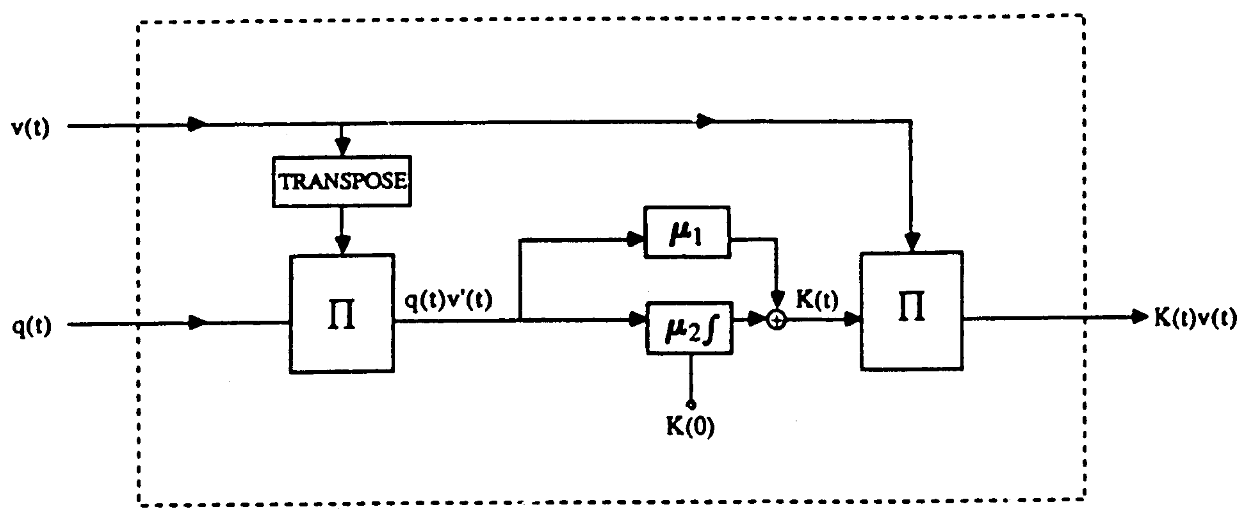


Figure 4. Structure of the Basic Adaptation Module

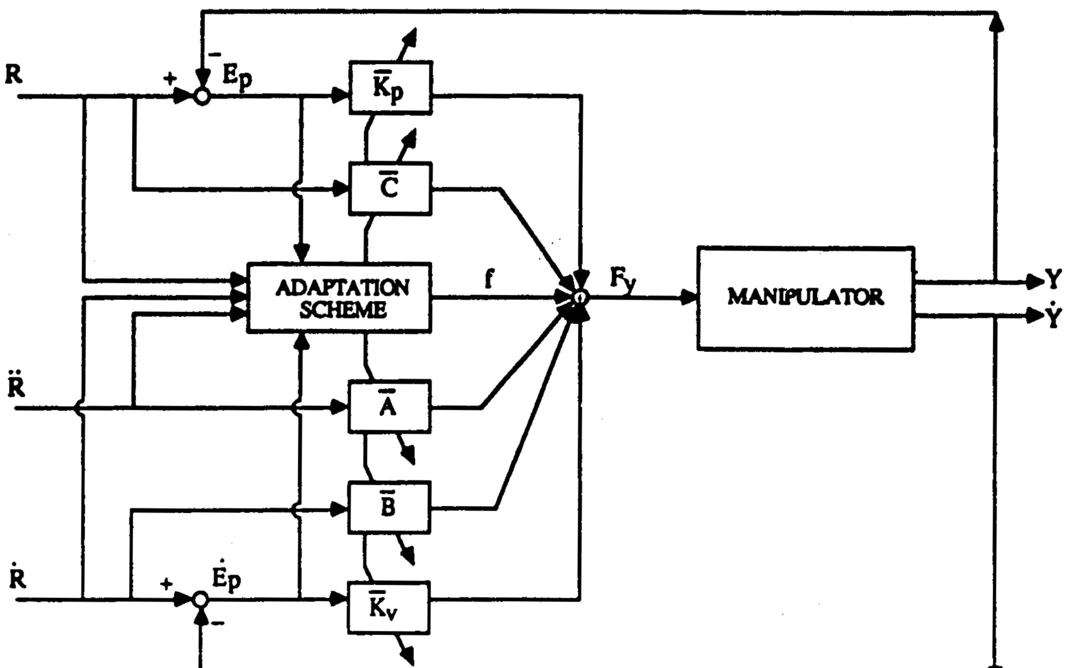


Figure 5. Adaptive Position Control System